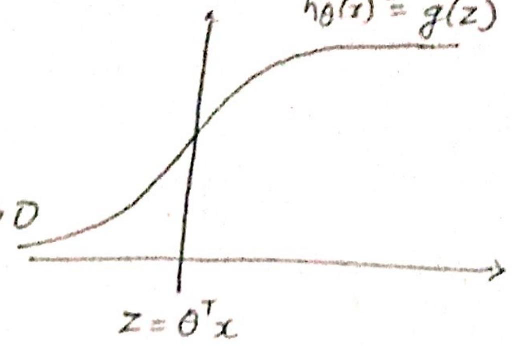
12.1 - Support Vector Machines | Optimization Objective

$$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$$



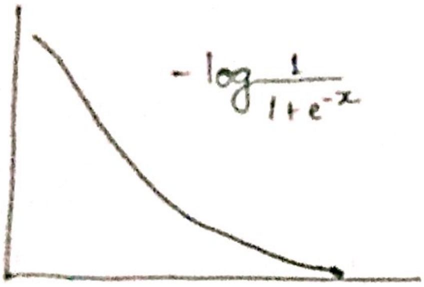
If $y = 1$, we want $h_\theta(x) \approx 1$, $\theta^T x \gg 0$

Alternative view of logistic regression

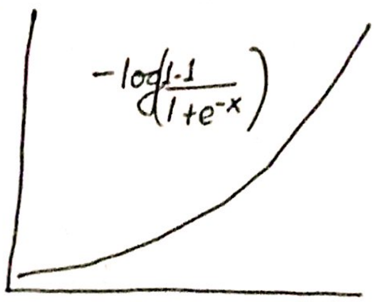
Cost of example: $-(y \log h_\theta(x) + (1-y)\log(1-h_\theta(x)))$

$$= -y \log \frac{1}{1+e^{-\theta^T x}} - (1-y)\log\left(1 - \frac{1}{1+e^{-\theta^T x}}\right)$$

If $y = 1$ (want $\theta^T x \gg 0$):



If $y = 0$ (want $\theta^T x \ll 0$):



logistic regression:

$$\min_\theta \frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)}\left(-\log h_\theta(x^{(i)})\right) + (1-y^{(i)})\left(-\log(1-h_\theta(x^{(i)}))\right)\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

Support vector machine:

$$\min_\theta C \sum_{i=1}^{m}\left[y^{(i)} \text{cost}_1(\theta^T x^{(i)}) + (1-y^{(i)}) \text{cost}_0(\theta^T x^{(i)})\right] + \frac{1}{2}\sum_{i=1}^{n}\theta_j^2$$

Hypothesis:

$$h_\theta(x) \begin{cases} 1 & \text{if } \theta^T x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

12.2: Large Margin intuition

If $y = 1$, we want $\theta^T x \geq 1$ (not just ≥ 0)

If $y = 0$, we want $\theta^T x \leq -1$ (not just < 0)

$$\min_{\theta} C \sum_{i=1}^{m} \left[y^{(i)} \text{cost}_1(\theta^T x^{(i)}) + (1-y^{(i)}) \text{cost}_0(\theta^T x^{(i)}) \right] + \frac{1}{2} \sum_{i=1}^{n} \theta_j^2$$

0

Whenever $y^{(i)} = 1$:
$\theta^T x^{(i)} \geq 1$

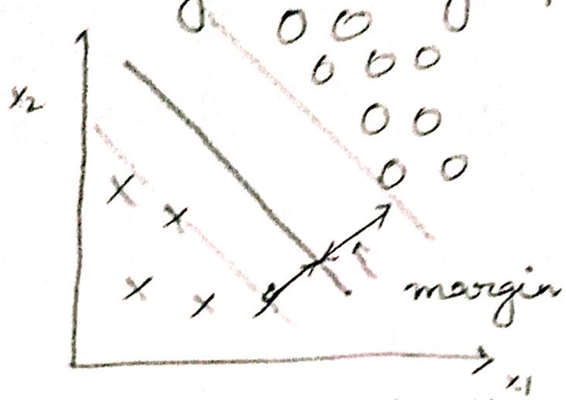
$\rightarrow \min_{\theta} C \times 0 + \frac{1}{2} \sum_{i=1}^{n} \theta_j^2$

Whenever $y^{(i)} = 0$:
$\theta^T x^{(i)} \leq -1$

s.t. $\theta^T x^{(i)} \geq 1$ if $y^{(i)} = 1$

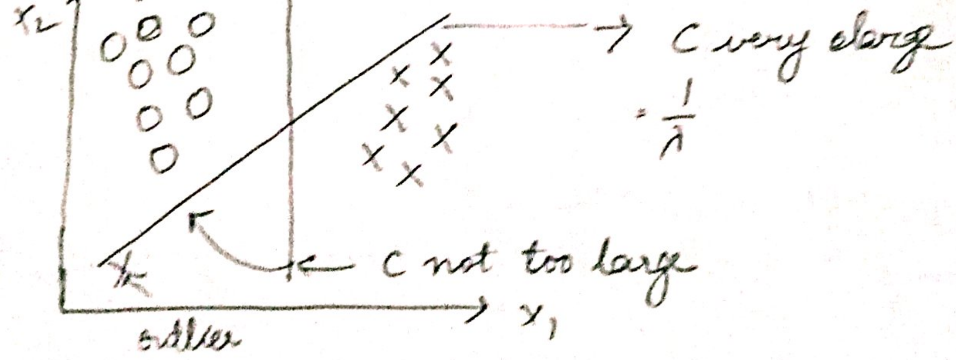
$\theta^T x^{(i)} \leq -1$ if $y^{(i)} = 0$

SVM Decision Boundary: Linearly separable case

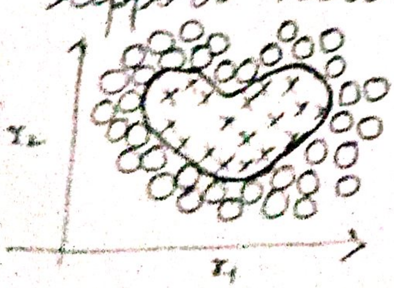


Large margin classifier

Large margin classifier in presence of outliers



12.4 - Support Vector Machines (Kernels-I)



Predict $y=1$ if
$\rightarrow \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \theta_4 x_1^2 + \theta_5 x_2^2 + \cdots \geq 0$

$$h_\theta(x) = \begin{cases} 1 & \text{if } \theta_0 + \theta_1 x_1 + \cdots \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

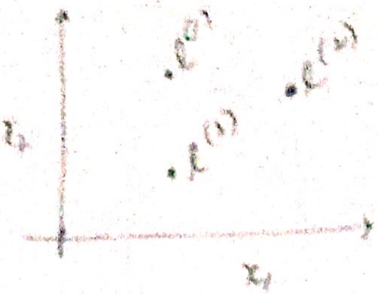
$\theta_0 + \theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3 + \cdots$

$f_1 = x_1$, $f_2 = x_2$, $f_3 = x_1 x_2$, $f_4 = x_1^3$, $f_5 = x_2^2$

Is there a better / different choice of the features $f_1, f_2, f_3, \ldots$?

Kernel

Given x, compute new features depending on proximity to landmark



Given x:

$f_1 = \text{similarity}(x, l^{(1)}) = \exp\left(-\frac{\|x - l^{(1)}\|^2}{2\sigma^2}\right)$

$f_2 = \text{similarity}(x, l^{(2)}) = \exp\left(-\frac{\|x - l^{(2)}\|^2}{2\sigma^2}\right)$

$f_3 = \text{similarity}(x, l^{(3)}) = \exp(\quad)$

kernel (Gaussian kernels)

Kernels and Similarity

$$f_1 = \text{similarity}(x, l^{(1)}) = \exp\left(-\frac{\|x - l^{(1)}\|^2}{2\sigma^2}\right) = \exp\left(-\frac{\sum_{j=1}^{n}(x_j - l_j^{(1)})^2}{2\sigma^2}\right)$$

If $x \approx l^{(1)}$:

$$f_1 \approx \exp\left(-\frac{0^2}{2\sigma^2}\right) \approx 1$$

If x is far from $l^{(1)}$

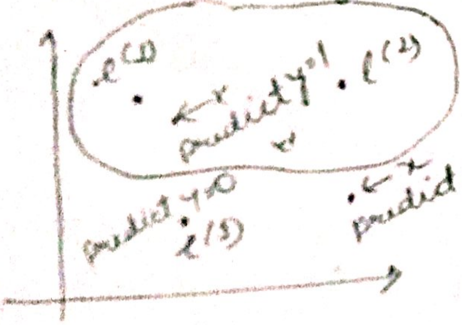
$$f_1 = \exp\left(-\frac{(\text{large Nos})^2}{2\sigma^2}\right) \approx 0.$$

$l^{(1)} \rightarrow f_1$

$l^{(2)} \rightarrow f_2$

$l^{(3)} \rightarrow f_3$

features



Predict "1" when

$\theta_0 + \theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3 \geq 0$

$\theta_0 = -0.5, \quad \theta_1 = 1, \quad \theta_2 = 1, \quad \theta_3 = 0$

$f_1 \approx 1, \quad f_2 \approx 0, \quad f_3 \approx 0$

$\theta_0 + \theta_1 \times 1 + \theta_2 \times 2 + \theta_3 \times 0$

$= -0.5 + 1 = 0.5 \geq 0$

$f_1, f_2, f_3 \approx 0$

$\rightarrow \theta_0 + \theta_1 f_1 + \ldots \approx -0.5 < 0$

12.5 – Support Vector Machines | Kernel-II

Choosing the landmarks

Where to get $l^{(1)}, l^{(2)}, l^{(3)}, \ldots$?

Given $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})$

choose $l^1 = x^{(1)}, l^2 = x^2, \ldots, l^{(m)} = x^{(m)}$

Given example x:

$f_1 = \text{similarity}(x, l^{(1)})$

$f_2 = \text{similarity}(x, l^{(2)})$

$f = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix}$ $f_0 = 1$

For training example $(x^{(i)}, y^{(i)})$

$x^{(i)} \rightarrow$

$f_1^{(i)} = \text{sim}(x^{(i)}, l^{(1)})$

$f_2^{(i)} = \text{sim}(x^{(i)}, l^{(2)})$

$f_3^{(i)} = \text{sim}(x^{(i)}, l^{(i)}) = \exp\left(\frac{-0}{2\sigma^2}\right) = 1$

$\vdots$

$f_m^{(i)} = \text{sim}(x^{(i)}, l^{(m)})$

$x^i \in \mathbb{R}^{n+1}$ or $\mathbb{R}^n$

$f^{(i)} = \begin{bmatrix} f_0^{(i)} \\ f_1^{(i)} \\ \vdots \\ f_n^{(i)} \end{bmatrix}$

$f_0^{(i)} = 1$

Hypothesis: Given x, compute features $f \in \mathbb{R}^{m+1}$

Predict "$y = 1$" if $\theta^T f \geq 0$

Training

$$\rightarrow \min_\theta C \sum_{i=1}^{m} y^{(i)} \text{cost}_1(\theta^T f^{(i)}) + (1 - y^{(i)}) \text{cost}_0(\theta^T f^{(i)}) + \frac{1}{2} \sum_{j=1}^{m} \theta_j^2$$

$C = \left(\frac{1}{\lambda}\right)$

Large C: Lower bias, high variance (small λ)

Small C: Higher bias, low variance (large λ)

σ^2

Large σ^2: Features f_i vary more smoothly.

Higher bias, lower variance

Smaller σ^2: Features f_i vary less smoothly

Lower bias, higher variance

12.6 – Using An SVM

Use SVM software package (e.g. liblinear, libsvm) to solve for parameter θ

Need to specify:

Choice of parameter C

Choice of kernels (similarity function)

E.g. No kernel ("linear kernel") $\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots \theta_n x_n \geq 0$

Predict "y=1" if $\theta^T x \geq 0$ → n large, m small $x \in \mathbb{R}^{n+1}$

Gaussian kernel:

$f_i = \exp\left(-\frac{\|x - l^{(i)}\|^2}{2\sigma^2}\right)$, where $l^{(i)} = x^{(i)}$ $x \in \mathbb{R}^n$, n small and/or m large

Need to choose σ^2

Kernel (similarity) functions:

function f = kernel (x1, x2)

$f = \exp\left(\frac{\|x1 - x2\|^2}{2\sigma^2}\right)$

return

Note: Do perform feature scaling before using the Gaussian model

Other choice of kernel

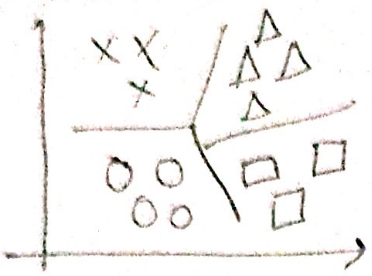
Note: Not all similarity functions similarity(x, l) makes valid kernels (Need to satisfy technical conditions called "Mercer's Theorem" to make sure SVM packages optimizations run correctly, and do not diverge).

Many off-the-shelf kernels available:

- Polynomial kernel:

- More esoteric: String kernel, chi-square kernel, histogram intersection kernels, ...

Multi Class classification



$y \in \{1, 2, 3, \ldots K\}$

Many SVM packages already have built-in-multi-class classification functionality

Otherwise, use one Vs all method. (Train K SVMs, one to distinguish $y = i$ from the rest, for $i = 1, 2, \ldots K$), gets $\theta^{(1)}, \theta^{(2)}, \ldots \theta^{(K)}$

Pick class i with largest $(\theta^{(i)})^T x$

Logistic regression VS SVMs

n = number of features ($x \in \mathbb{R}^{n+1}$), m = number of training examples

If n is large (relative to m): E.g. $n \geq m$, $n = 10{,}000$, $m = 10 \ldots 1000$)

Use logistic regression, or SVM without kernel ("linear kernel")

If n is small, m is intermediate:
→ Use SVM with Gaussian kernel

If n is small, m is large
→ Create/add more features, then use logistic regression or SVM without a kernel

→ Neural network likely to work well for most of these setting, but may be slower to train